𝓃

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/768,482	01/23/2001	Greg Wiggins	47578.0100	1535

7590 12/06/2006

John R. Thompson, Esq.
STOEL RIVES L.L.P.
Suite 1100
201 South Main Street
Salt Lake City, UT 84111

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 12/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/768,482	WIGGINS ET AL.
	Examiner	Art Unit
	Ted T. Vo	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 14 August 2006.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1,3-7 and 10-18 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1,3-7 and 10-18 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.	6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed on 08/14/2006.

Claims 1, 3-7, 10-18 remain pending.

Response to Arguments

2. Claim 3 is amended. The allowable subject matter to this claim is withdrawn.

Applicants' argument in the remarks filed on 08/14/06 has been considered, but not persuasive. See definition of CONSOLE from http://www.webopedia.com/TERM/C/console.html:

(1) The combination of display monitor and keyboard (or other device that allows input). Another term for console is *terminal*. The term console usually refers to a terminal attached to a minicomputer or mainframe and used to monitor the status of the system.

As known, a WINZIP resides in a Microsoft Windows, the WINZIP itself is a console because it is a software module and it allows a user to build a package, that is a .ZIP file. Simply, double click on an executable icon of WINZIP, or run WinZIP, a user can retrieve a WINZIP as shown by the reference, it reads means of console: "Create self-extracting ZIP files With WinZip 6.3". What are the differences to this WINZIP to the Applicants' claims? It should be known the WINZIP is communicating with the Windows' registry. A user builds a self-extracting .ZIP file (the package) very easy if he/she knows how to use the Windows operating system.

Microsoft Windows is a console. MSDOS commands used in Windows operating system are known as Windows Console commands. The Microsoft Windows that has the WINZIP clearly performs all of the functions read from the claims.

See Link: http://effbot.org/downloads/index.cgi/console-1.1a1-20011229.zip/module-console.html

The Console Module

[This is work in progress. Last updated December 2, 2000]

The *Console* module provides a simple console interface, which provides cursor-addressable text output, plus support for keyboard and mouse input.

The *Console* module is currently only available for Windows 95, 98, NT, and 2000. It probably works under Windows XP, but it hasn't been tested on that platform.

Software (including precompiled binaries) and documentation can be downloaded from http://effbot.org/efflib/console

Concepts

Screen

The console screen consists of a 2-dimensional grid containing character cells. All characters cells have the same size.

Each character cell has a unique coordinates. The coordinate origin (0, 0) is in the upper left corner, as usual.

You can use negative coordinates as well. They work pretty much like negative sequence indexes in Python; for example, column -1 is the rightmost column on the console.

The **rectangle** and **save** methods require you to specify character rectangles. A rectangle is a tuple containing two coordinate pairs. The second pair of coordinates specify the cell just to the right and below the last cell in the rectangle. In other words, the rectangle (0, 0, 20, 10) is twenty characters wide and ten characters high.

Note that the second coordinate must be equal to or larger than the first coordinate. Also, the current implementation doesn't support negative coordinates in rectangles.

Cursor

The console driver keeps track of the current cursor position. However, only three methods actually use this position: **write**, **page**, and of course **pos**.

The **cursor** method allows you to switch the cursor on and off.

Styles

Several methods take **style** arguments. The style is an integer value that combines a foreground and a background color. The console driver supports 16 colors in total:

```
0 = black (#000000)
1 = blue (#0000a8)
2 = dark green (#00a800)
3 = n/a (#00a8a8)
4 = red (#a80000)
5 = magenta (#a800a8)
```

```
6 = brown (#a8a800)
7 = light grey (#a8a8a8)
8 = dark grey (#545454)
9 = n/a (#5454fc)
10 = green (#54fc54)
11 = cyan (#54fcfc)
12 = n/a (#fc5454)
13 = n/a (#fc54fc)
14 = yellow (#fcfc54)
15 = white (#fcfcfc)
```

To calculate the style, combine the colour for the foreground and background as follows:

```
style = foreground + background*16
```

The default style is light grey on black background.

TBD: add names for remaining colors (see the HTML 3.2 specification?)

The Console Class

To create a **Console** instance, import the **Console** module and call the **getconsole** factory function.

Example: Example: Using the Console module

```
import Console

c = Console.getconsole()

c.title("Console Example")

c.text(0, 0, "here's some white text on white background", 0x1f)
c.text(10, 5, "line five, column ten")
```

Console Methods

Instances returned by **getconsole** have the following methods:

Basic Graphic Methods

text

text(column, line, string, style)

Write **string** to the screen at the given position, using the given **style**. This method does not move the cursor. If the style is omitted, it defaults to white text on black background.

rectangle

rectangle(rect, style, character)

Blanks the given rectangle. The **rect** argument should be a 4-tuple. If the **character** argument is omitted, it defaults to space. If the **style** argument is omitted, it defaults to black.

scroll

scroll(rect, dx, dy, style, character)

Moves the given rectangle **dx** cells to the right (or left, if dx is negative), and **dy** cells down (or up). The **style** and **character** attributes are used to fill empty regions. If the **character** argument is omitted, it defaults to space. If the **style** argument is omitted, it defaults to black.

page

page()

Blanks the screen, and moves the cursor to (0, 0).

page

page(style, character)

Clears the screen, using the given style and fill character. The style will also become the new default style. It moves the cursor to (0, 0).

File-Like Output

write

write(string)

Writes **string** at the current position, using a default style. This method treats the console as a conventional text terminal, which means that tabs, backspace, newline, and the bell character works as expected. The cursor is moved to the position just after the last written character.

pos

pos(column, line)

Moves the cursor to the given column and line. If the coordinates are omitted, this method returns the current cursor position [TBD: better name?]

softspace

softspace *(integer def)*

This attribute isn't used by the console driver itself, but is there to make sure Python's **print** statement works as expected when printing to a console device.

Input

get

get() => *event*

Get the first event from the input queue. The return value is an instance of the Event class. If the input queue is empty, this method blocks.

getchar

getchar() => *event*

Get the first character event from the input queue, ignoring any other kind of event. The return value is an instance of the KeyPress class. If the input queue is empty, this method blocks.

peek

peek() => *event*

Fetch the first event from the input queue, without removing it. If the input queue is empty, this method returns None.

Console Properties

cursor

cursor(flag)

cursor(0) makes the text cursor invisible. **cursor(1)** makes it visible. [TBD: better name?]

size

size() => *(width, height)*

Get the current size of the console window, as a 2-tuple (width, height).

title

title(string)

Set the window title to **string**. If the string is omitted, this method returns the current title.

save

save()

TBD: document save and restore.

The Event Class

The **get** and peek methods return event descriptors. These are modelled after the Tkinter **Event** type, with a few minor differences.

Event Subtypes

Note: In the current release, all event objects are implemented using the same Python type. To check what kind of event you have, use the type attribute.

KeyPress

KeyRelease

ButtonPress

ButtonRelease

Motion

Configure

Event Attributes

type

type *(string)*

The event type, given as a string. This attribute contains one of **KeyPress**, **KeyRelease**, **ButtonPress**, **ButtonRelease**, **Motion**, or **Configure**. Other event types may be returned, but such events should always be ignored by the application.

char

char *(string)*

The character code for **KeyPress** or **KeyRelease** events. If this is empty, the event doesn't have an ASCII representation.

keycode

keycode *(integer)*

The numerical keycode for **KeyPress** or **KeyRelease** events.

keysym

keysym *(string)*

The symbolic name for **KeyPress** or **KeyRelease** events. If this is an empty string, this event doesn't have a symbolic name.

state

state *(integer)*

The button and control key state. This is valid for **KeyPress**, **KeyRelease**, **ButtonPress**, **ButtonRelease**, and **Motion** events.

x, y

x *(integer)*

y *(integer)*

The current mouse coordinate. This is valid for ButtonPress, ButtonRelease, and Motion events.

width, height

width *(integer)*

height *(integer)*

The new window size. This is only valid for Configure events.

serial

serial *(integer)*

serial is the serial number for this event.

time

time *(integer)*

The relative time in milliseconds when this event was generated. In the current implementation, this is always 0.

widget

widget *(widget or None)*

The widget that generated this event. Unless you're using **uiToolkit/Text** or another console widget toolkit built on top of this module, this is always **None**.

License

The *Console* module was written by Fredrik Lundh at Secret Labs AB, in January 1999.

Therefore, Microsoft Windows is a console; it has a console module. MSDOS commands used in Windows operating system are known as Windows Console commands. The Microsoft Windows that has the WINZIP clearly performs all of the functions read from the claims. So does the reference.

Specification

3. This Application contains continuation data, claimed benefit under 60/177,585. It should require to include explicitly with **incorporation by reference** statement, pursuant to 37 CFR 1.57: Incorporation by reference.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1, 4-7, 10-17 are rejected under 35 U.S.C. 102(b) as being anticipated by Nico Mak, WinZip Version 7.0.

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per Claim 1: The WinZip application is used for generated a .ZIP file (Note: Version 6.3 of the Inside the Internet is obvious before Version 7.0, and before the effective filing date of this application). Nico Mak discloses a system for extracting application information, included with an application WinZip registered in a Microsoft Window. The WinZip associates with Microsoft Windows' console to provide a user this application to create a compresses package ".ZIP file". Within a created .ZIP file, it packs the files in which a user manages his files on his own computer or sends to the other users who want to share the files. The .ZIP is compressed; therefore it migrates easily to other computers in the Internet. Nico Mak, thus discloses, "*A system for describing and extracting application information, comprising:*

(A) a first computer system, said first computer system further comprising:(1) an input device; (2) a display device; and (3) a processing unit, said processing unit further comprising:(a) a processor;(b) memory; and(c) a long-term storage device; (computer per se)

(B) a second computer system, said second computer system further comprising:(1) an input device; (2) a display device; and (3) a processing unit, said processing unit further comprising:(a) a processor;(b) memory; and(c) a long-term storage device; (computer per se)

(C) an application program resident on said long term storage device of said processing unit of said first computer system (Computer provides storage for storing applications); and

(D) a means for migrating said application program from said first computer to said second computer, (connecting and communication via Internet: see http://www.winzip.com, p.1) *where means for migrating further comprises:*

(1) an application interface (See WinZIP 7.0 associated with Windows NT, used to create a .ZIP file), where this reference uses a Microsoft Windows.

(2) a communication channel between said first computer system and said second computer system (Refer to Internet connected between two computers);

(3) a console in communication with said application interface file (See page 1-2, start at "View Dialog box", and all commands that has means for creating a .Zip, such as File, Add, Drop; etc. Furthermore, the Windows with its basics windows' commands and Windows registry, in which the WinZip is registered, is also a console); *and*

(4) a self-extracting auto-migration package built by said console said self-extracting auto-migration package further comprising files and self-extracting auto-migration package further comprising files and settings for migration to said second computer (That is any .ZIP file created by WinZip 7.0).

As per Claim 4: Nico Mak discloses the WinZip with Drag and Drop, "File Properties" for creating self-extractor packages.

As per Claim 5: Nico Mak discloses the WinZip which is adaptable to a standard Window like Window 95, editable by a "File Properties".

As per Claim 6: With regard to limitation of Claim 6, see all commands shown in page 2.

As per Claim 7: With regard to limitation of Claim 7, Self-Extracting Zip file is an executable file.

As per Claim 10: With regard to limitation of Claim 10, see page 2, "move", "Add", "Drop", "File Properties", allowing a user to edit a file.

As per Claim 11: With regard to limitation of Claim 11, see page 13, Winzip.com provides password to it customer.

As per Claim 12: With regard to limitation of Claim 12, ZIP file is a compressed package.

As per Claim 13: With regard to limitation of Claim 13, Buttons EXTRACT in the ZIP file.

As per Claim 14: With regard to limitation of Claim 14, associated with Windows commands.

As per Claim 15: With regard to limitation of Claim 15, the text area in any WinZip file. It should be noted that it is adaptable from Microsoft Windows feature.

As per Claim 16: With regard to limitation of Claim 16, commands such as "Add", "Drop", "Clipboard Copy", allow a file to be copied and migrated in the Self-Extractor or deleted when they are duplicated. Furthermore, this reference is associated with Microsoft Windows that is included with Windows console commands, i.e., when drop a file in to a directory or a text area or a .ZIP file, if the drop file is in there, Windows console command always comes up with a warning question.

As per Claim 17: With regard to limitation of Claim 17, see means of self-extracting.

6. Claims 1, 3-7, 10-18 are rejected under 35 U.S.C. 102(a) as being anticipated by Network Associates, Inc., "Total Virus Defense Suite – Best Practices Guide - Migration Technical Manual" Version 4.x family products, pages: i-x, 1-248, 5-1999 (hereinafter: NA).

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per Claim 1: NA discloses a system for extracting application information. NA utilizes Microsoft Windows (p. 50) and common Self Extracting Application (p.53) to migrate its files to Network users. The .ZIP is compressed; therefore it is easily downloaded by the Network users. Thus NA discloses,

"*A system for describing and extracting application information, comprising* (Version 4.x family products including the Networks users who utilize Microsoft Windows):

(A) a first computer system, said first computer system further comprising:(1) an input device; (2) a display device; and (3) a processing unit, said processing unit further comprising:(a) a processor;(b) memory; and(c) a long-term storage device (computer per se, p. 54 or p. 196: refer to the showing of any computer, a client, or a server);

(B) a second computer system, said second computer system further comprising:(1) an input device; (2) a display device; and (3) a processing unit, said processing unit further comprising:(a) a processor;(b) memory; and(c) a long-term storage device (computer per se, p. 54 or p. 196: refer to the showing of any computer, a client, or a server);

(C) an application program resident on said long term storage device of said processing unit of said first computer system (Computer provides storage for storing applications); and

(D) a means for migrating said application program from said first computer to said second computer (Computer factory provides modem and input/output port for connecting and communication with other computers: See in the back of a computer and see circuit board/card/modem at a standard computer – herein NA provide a migration via network) .

where means for migrating further comprises:

(1) an application interface (See interface/functionality technology included in the product, see in table 2-2, started at p. 16);

(2) a communication channel between said first computer system and said second computer system (See command line scanner, started at p. 21, used in Windows for downloading);

(3) a console in communication with said application file (The reference Guide is a console, which has "Main console" (p. 39) can communicate with the file system/directory of the Windows in which it resides (See all tables and Key settings start at p. 123); *and*

(4) a self-extracting auto-migration package built by said console (The reference has functionality to build a self-extracting package from its main console, e.g. see table 2-1, p. 12. It should be noted that the Windows is a console because it allow to built a file and attached the file in its windows), *said self-extracting auto-migration package further comprising files and self-extracting auto-migration package further comprising files and settings for migration to said second computer* (See p. 54, the main console provides interface, auto update, schedule and send to a client a zip of files: The files in the ZIP could be DAT files or other files shown in table 2-1 (p. 12), i.e. '*comprising files*', and key options (p. 39) such as AVCONSOL.INI file '*comprising settings*', which is available for installation pre-configuration – further see p. 42, Migrating from Virusscan 3.X, or from Management Edition, etc.).

As per Claim 4: NA discloses, modules of Windows that is able to scan a ZIP application to create a .ZIP file of .DAT files. A user uses main console (p. 54) to create migration packages.

As per Claim 5: NA discloses, a standard Windows with "EDIT" key in registry setting (p.131). It should be known that all Microsoft commands are Windows Console commands. EDIT key is used not only in WINDOWS, but also in ZIP application.

As per Claim 6: With regard to limitation of Claim 6, that is ZIP application included within Microsoft Windows within this NA. For example, the Registry settings included configFile (p.131).

As per Claim 7: With regard to limitation of Claim 7, see Zip file.

As per Claim 10: With regard to limitation of Claim 10, Windows, or ZIP file is editable file.

As per Claim 11: With regard to limitation of Claim 11, e.g., see page 238, "PASSWORD".

As per Claim 12: With regard to limitation of Claim 12, i.e., the meaning of ZIP file.

As per Claim 13: With regard to limitation of Claim 13, i.e., Microsoft Windows Console commands.

As per Claim 14: With regard to limitation of Claim 14, i.e., boot, reboot, configFile.

As per Claim 15: With regard to limitation of Claim 15, i.e., such text such as name, date, time etc., of a file in the WinZip, or shown by Windows.

As per Claim 16: With regard to limitation of Claim 16, Microsoft Windows console command that is utilized in every associated application. e.g. when drop a file in to a directory or a text area or a .ZIP file, if the drop file is in there, Windows console command always comes up with a warning question.

As per Claim 17: With regard to limitation of Claim 17, see means of self-extracting.

As per Claim 18: With regard to limitation of Claim 18, NA show table E-9, with profile information, this would be included in the .ZIP file.

As per claim 3:

NA discloses:

(A) loading a Personality Package, said Personality Package further comprising setting, preferences, application programs, and data files;

(B) executing said Personality Package;

because it includes a package deployed using operating system such as Windows NT and 9x, Tivoli, and others, where with command line, each of this operating system can load and execute the deployed package. See page 32 and up, execute a package such as SMS package.

(C) getting a file; (D) copying said file; and determining whether migration of additional files is required and if additional files are required returning to said getting a file step; A user manually performs all acts, and thus NA discloses these limitations. Furthermore, NA discloses Windows NT, 3x, 9x and other Operating systems. Through Windows' commands, a user can get a file and copy the file. In p. 92, Windows 9x can copy .DAT files. Further see p. 31, NA shows feature, "Product review and evaluation" for reviewing documents required for migration – Further see p. 102, a Zip of DAT files will be configured and copied into a local machine key of a registry. Also see un-zipped DAT files.

(E) getting a registry; (F) copying said registry; A user manually performs all acts, and thus the NA discloses these limitations. Furthermore, NA discloses, registry settings, therefore, user of NA use Microsoft Windows Console commands to perform (E) and (F). NA further shows a script, started at p. 98, that can get a registry and copy registry: refer to command setRegValue ("HKEY_LOCAL_MACHINE") in Windows NT.

(G) getting application version specifics, and testing to determine if destination application version match and generating an error if said destination application versions do not match; and (H) updating links

With regards to above limitation: First of all the testing for migration is provided with Management Edition (p. 32). Secondly, NA includes auto update/upgrade (p. 54-55) that scan a registry, self-extract files in a ZIP file. NA provides "Registry settings" included with updating versions (See table E-6), and shows an .INI file with link tag: InstallDir (p.222) and vision identifier tag: Version (p. 223). A script in page 221 shows code that performs configuring version match and checking registry. NA shows script commands in p. 229-230, such as FILEEXISTS used to test the path of a file in installation, COPY, MOVE, DELETE, RENAME, used to update file path of a file in installation.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claim 18 is are rejected under 35 U.S.C. 103(a) as being unpatentable over Nico Mak, WinZip Version 7.0.

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per Claim 18: Nico Mak does not express show Self-extracting auto-migration package further comprises "*user-defined profiles*"

Official notice is taken that User-defined profiles recited in the claim is only data limitation that is included but does nothing in the claim.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of invention was made to include information file, data file, as part of compressed files in a package.

Conclusion

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 5:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number **571-273-8300**.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV
October 27, 2006

[illegible]

TED VO
PRIMARY EXAMINER
TECHN[illegible]Y CENTER 2100